Optimal Blocking by Minimizing the Maximum Within-block Distance

Michael J. Higgins[1]
Jasjeet Sekhon[2]

1) Princeton University
2) University of California at Berkeley

August 05, 2013

Funding generously provided by the San Francisco Bay Area Chapter of the ASA

Goal

Our goal:

- Devise a blocking method for experiments that ensures good covariate balance between treatment groups.
- Allow for designs with multiple treatment categories and multiple replications of each treatment within each block.
- Efficient enough to be used in large experiments (hundreds of thousands to millions of observations).

Covariate imbalance in randomized experiments

- PROBLEM: In randomized controlled experiments, depending on method of treatment assignment, there may be a non-negligible probability of bad balance on an important covariate between treatment groups—a treatment group has too many Republicans, severely ill people, etc.
- Bad balance on important covariates $\longrightarrow$ Imprecise estimates of treatment effects.
- Most likely to happen in small experiments; critical to avoid if cost of additional units is large (e.g. a medical trial).
- Better if randomization ensures good covariate balance instead of having to adjust for imbalance after treatment assignment.

Blocking to ensure covariate balance

Blocking can reduce variance of treatment effect estimates AND can be designed so that randomization preserves covariate balance.

We analyze the following blocking method:

1. Choose a measure of distance (e.g. Mahalanobis distance) that is small when important covariates have similar values.
2. Choose a threshold t^* for the minimum number of units to be contained in a block.
3. Obtain blocks: Each block contains at least t^* units and the maximum distance between any two units within a block—the maximum within-block distance (MWBD)—is minimized.

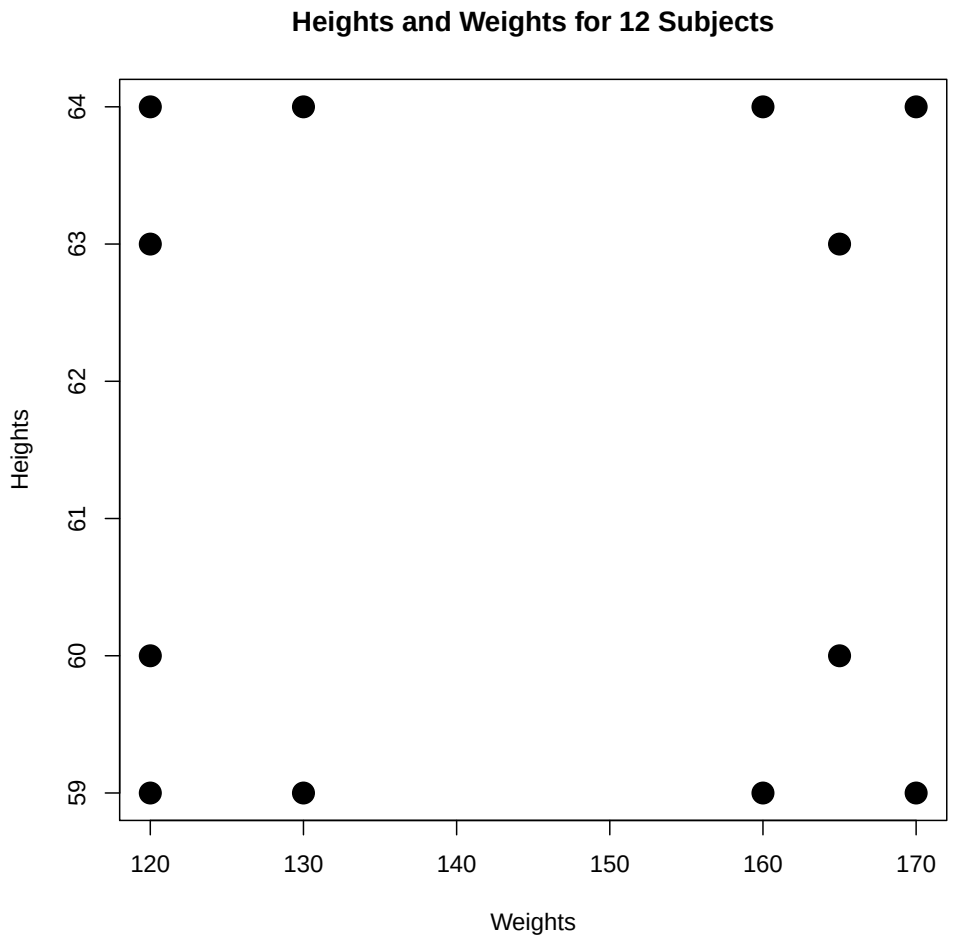
Blocking by minimizing the MWBD

Obtain blocks: Each block contains at least t^* units and the maximum distance between any two units within a block—the maximum within-block distance (MWBD)—is minimized.

- Minimizing the MWBD: Can ensure covariate balance in randomization.
- Threshold t^*: Allows designs with multiple treatment categories, multiple replications of treatments within a block; blocks can preserve clustering in data.
- "Good" blocking can be found in polynomial time: applicable to large experiments.

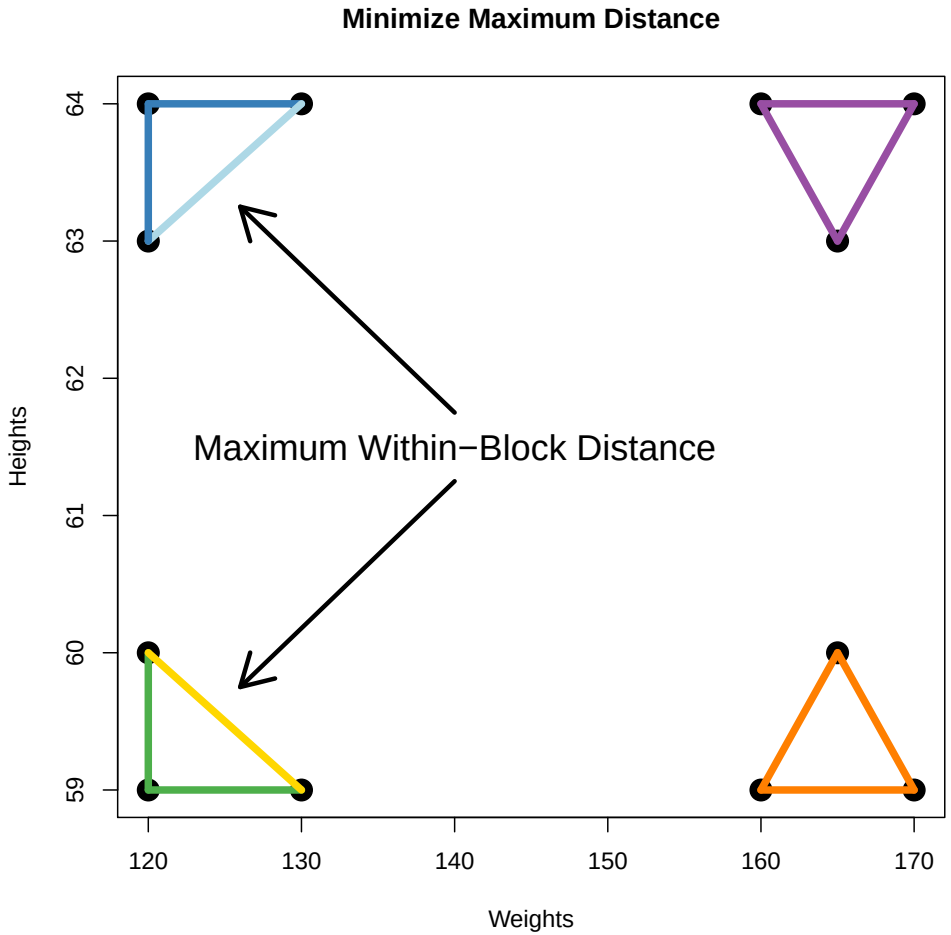
A simple example:

Threshold $t^* = 2$. Distance = Mahalanobis distance.



A simple example:

Threshold $t^* = 2$. Distance = Mahalanobis distance.



Optimal blocking and approximately optimal blocking

- *For all blockings that contain at least t^* units with each block*:
- Let λ denote the smallest MWBD achievable by such a blocking—any blocking that meets this bound is called an *optimal blocking*.
- Finding an optimal blocking is an NP-hard problem—feasible to find in small experiments, may not be feasible in large experiments [Hochbaum and Shmoys, 1986].

Optimal blocking and approximately optimal blocking

- *For all blockings that contain at least t^* units with each block*:
- Let λ denote the smallest MWBD achievable by such a blocking—any blocking that meets this bound is called an *optimal blocking*.
- Finding an optimal blocking is an NP-hard problem—feasible to find in small experiments, may not be feasible in large experiments [Hochbaum and Shmoys, 1986].
- We show that finding a blocking with MWBD $\leq 4\lambda$ is possible in polynomial time—finds a "good" blocking when the number of units is small *or large*.
- Denote any such blocking as an *approximately optimal blocking*.

Viewing experimental units as a graph

- Extending an idea from Paul Rosenbaum [1989]: Statistical blocking problems can be viewed as graph theory partitioning problems.
- Experimental units are vertices in a graph.
- An edge is drawn between two units if they can be placed in the same block.
- Each edge has a corresponding distance that is small if pretreatment covariates are similar (e.g. Mahalanobis distance).
- Use methods in graph theory to solve original blocking problem.

Notation:

- A graph G is defined by its vertex set V and its edge set E: $G = (V, E)$.
- Vertices in V denoted by $\{i\}$; n units $\longrightarrow$ n vertices in V.
- Edges in E are denoted by (i, j): at most $\frac{n(n-1)}{2}$ edges.
- The distance of edge $(i, j) \in E$ is denoted by d_{ij}:
  Only source of information about values of covariates.

Notation:

- Distances satisfy the triangle inequality: for any distinct vertices $\{i\}, \{j\}, \{k\}$,

$$d_{ik} + d_{kj} \geq d_{ij}.$$

- A *partition* of V is a division of V into disjoint *blocks* of vertices $(V_1, V_2, \ldots, V_\ell)$.
  Each vertex in V is assigned to exactly one block V_j.
- Blocking of units $\leftrightarrow$ Partition of a graph:
  If vertices are in the same block of a partition, the corresponding experimental units are in the same block.

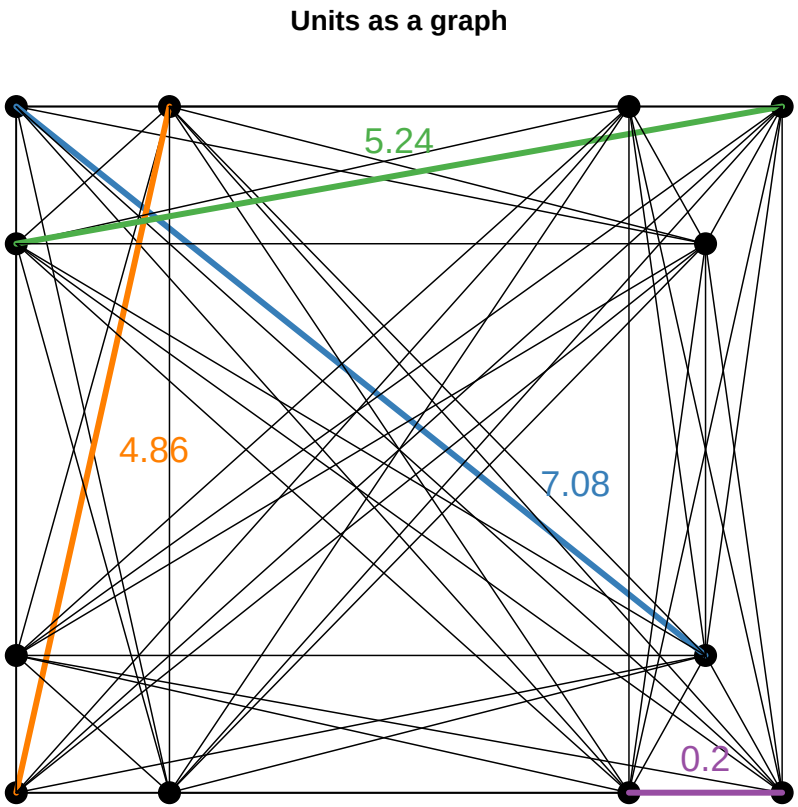
Bottleneck subgraphs

Large literature about graph partitioning problems.
Our primary tool: **Bottleneck subgraphs.**

- Define the *bottleneck subgraph with distance threshold d* as the subgraph that draws an edge (i, j) if and only if that edge has distance $d_{ij} \leq d$.
- Connect points in the same block through a path of edges within the bottleneck subgraph.
  With triangle inequality, obtain approximate optimality.

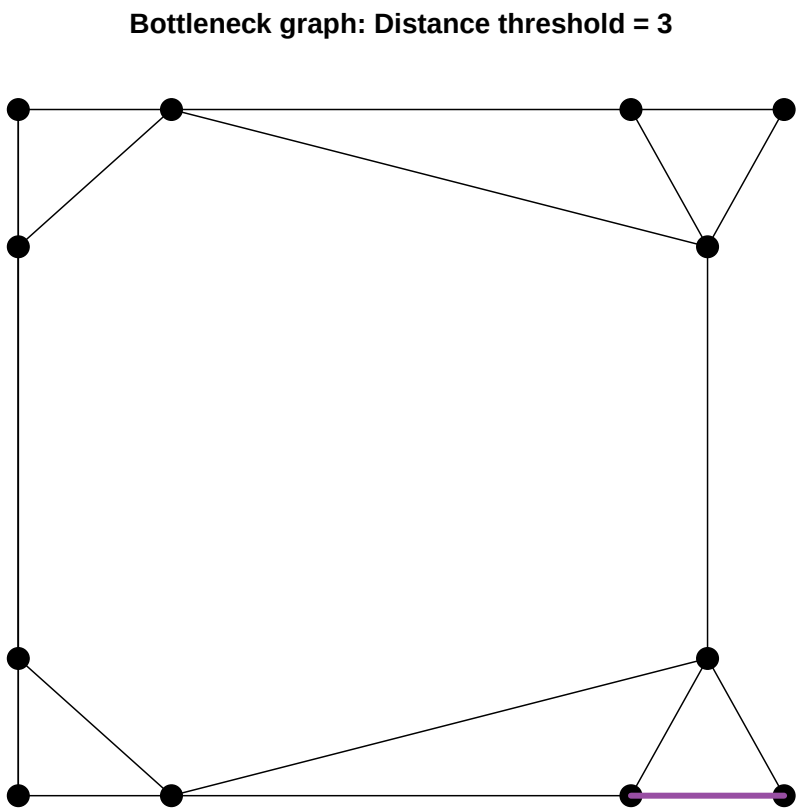
Bottleneck subgraph: In pictures

Complete graph



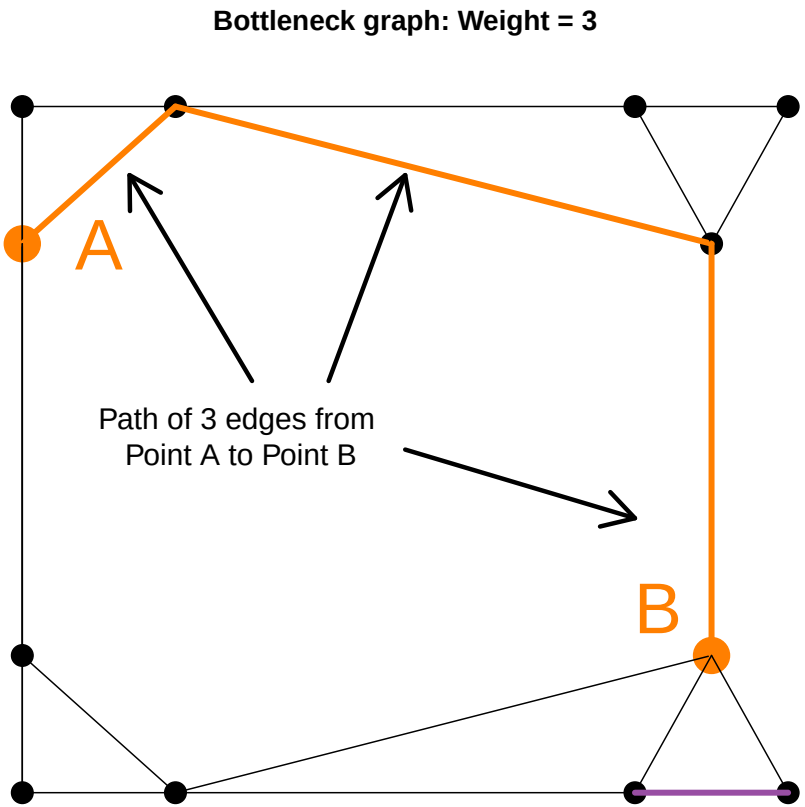
Bottleneck subgraph: In pictures

Bottleneck subgraph with distance threshold 3



Bottleneck subgraph: In pictures

Bottleneck subgraph with distance threshold 3



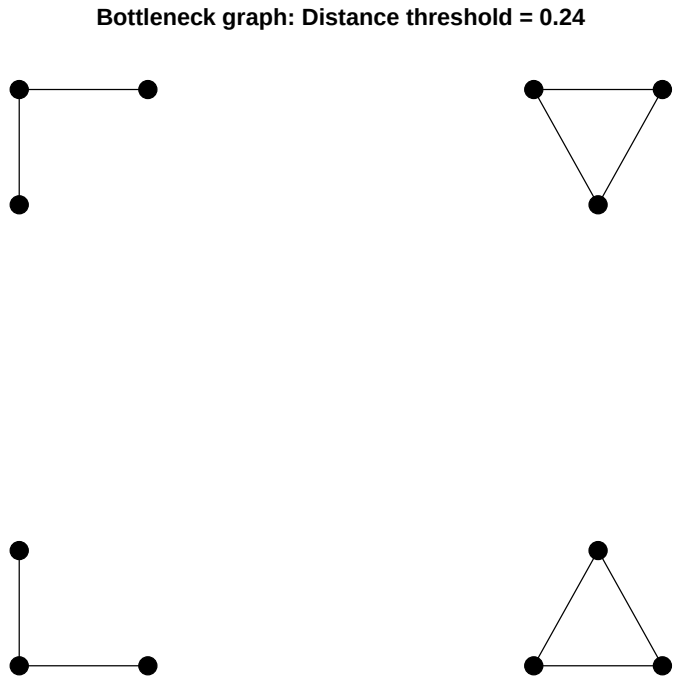
Approximate algorithm outline:

- Find a "sufficiently small" bottleneck subgraph.
- Select a set block centers that are "just far enough apart."
- Grow from these block centers within the bottleneck subgraph to obtain an approximately optimal partition—and thus, an approximately optimal blocking.
- Algorithm does not contain any inherently random components.
- Approach closely follows Hochbaum and Shmoys [1986].

Algorithm step-by-step: Find bottleneck graph

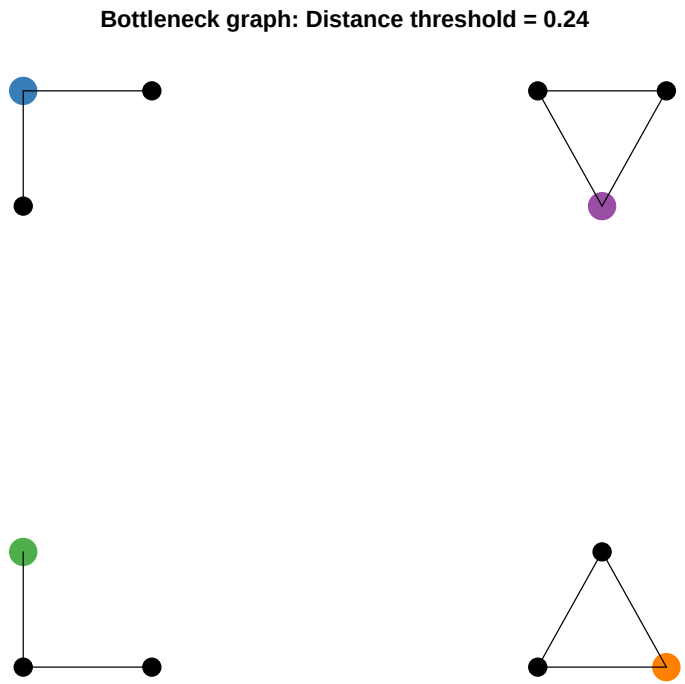
- Find the smallest distance threshold λ^- such that each vertex in the corresponding bottleneck subgraph is connected to at least $t^* - 1$ edges.
- Can show that $\lambda^- \leq \lambda$, where λ is the smallest MWBD possible.
- Subgraph can be constructed in polynomial time

$$t^* = 2$$



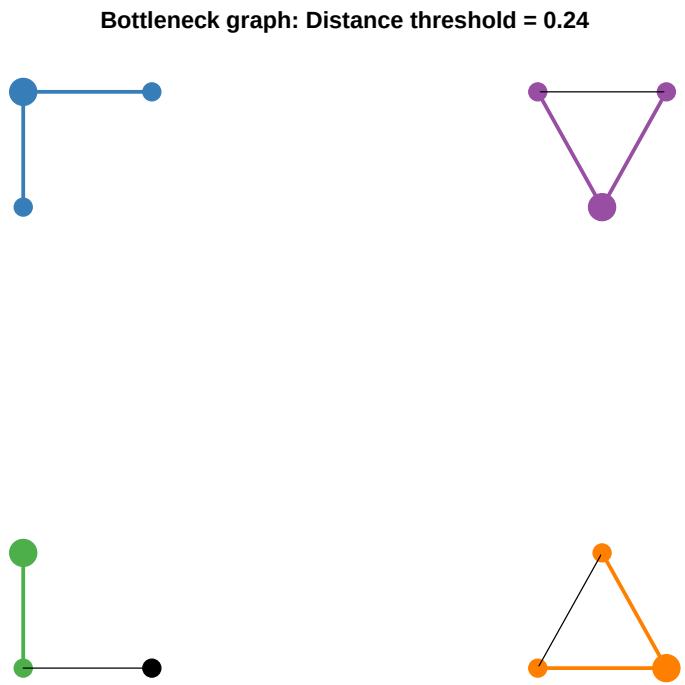
Algorithm step-by-step: Find block centers

- Find a set of vertices—*block centers*—such that:
  1. There is no path of two edges or less connecting any of the vertices in the set.
  2. There is a path of two edges or less that connects any vertex not in the set to one that is.
- Any set will do, but some choices of centers are better.



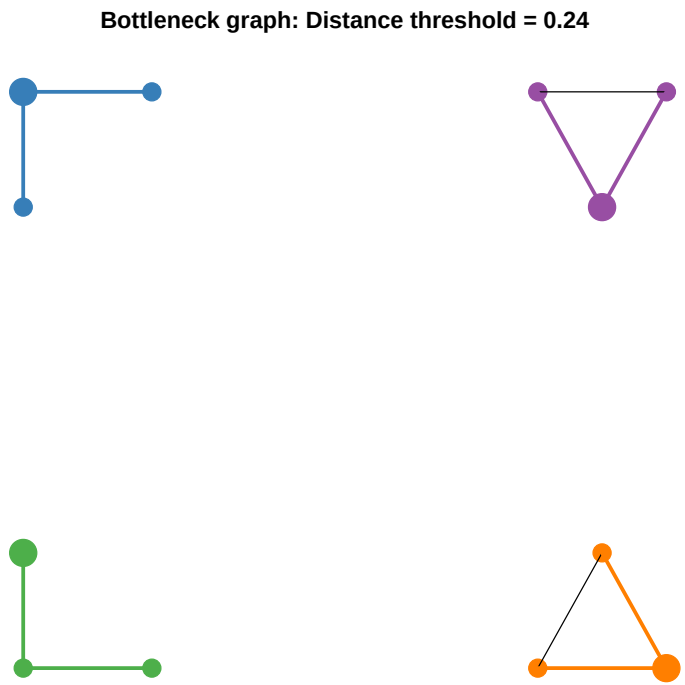
Algorithm step-by-step: Grow from block centers

- Form blocks comprised of a block center plus any vertices connected to that center by a single edge.
- Smart choice of block centers = these blocks will not overlap.
- These blocks contain at least t^* units (by choice of bottleneck subgraph).



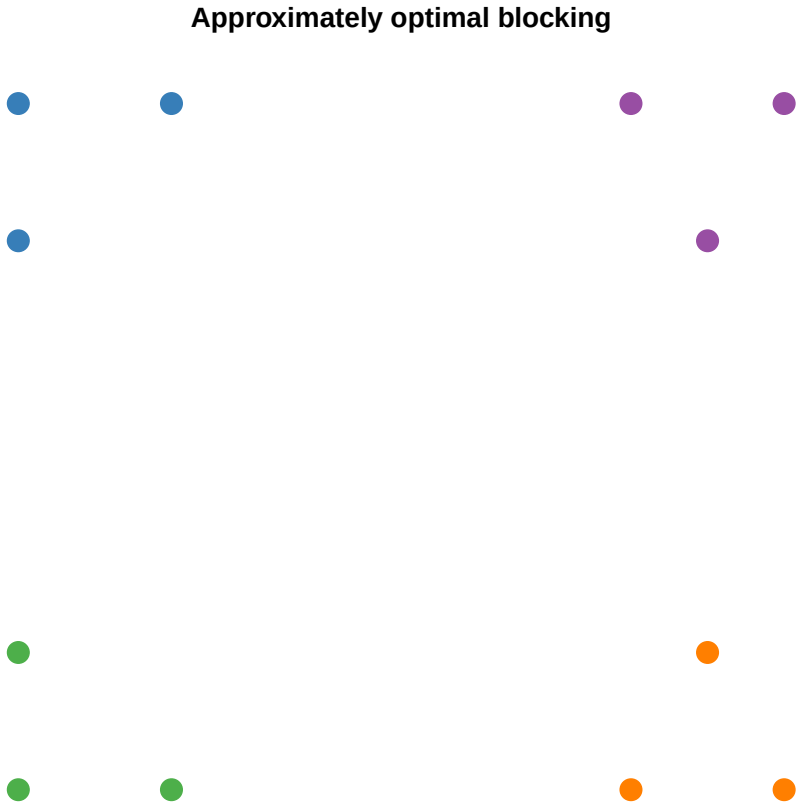
Algorithm step-by-step: Assign all unassigned vertices

- For each unassigned vertex, find the closest block center. Add that vertex to the center's corresponding block.
- Smart choice of block centers = all unassigned vertices are at most a path of two edges away from a block center.



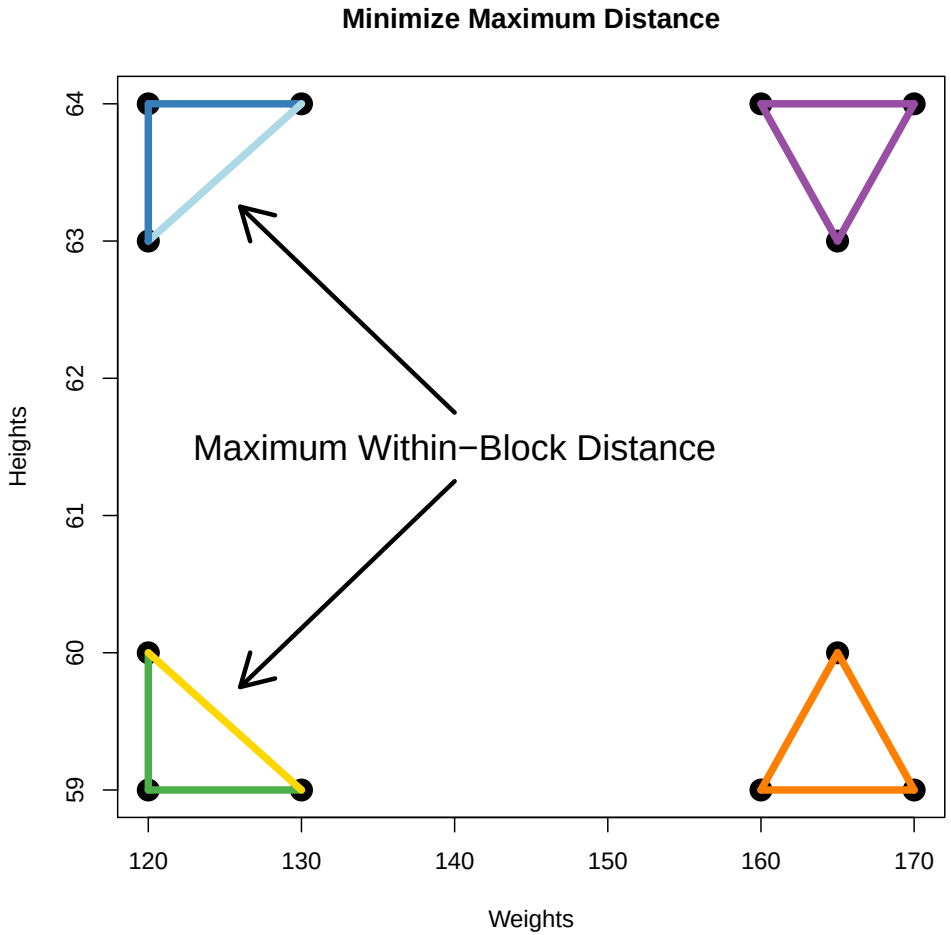
Our blocking

Our approximate algorithm came up with the following blocking:



A simple example:

Threshold $t^* = 2$. Distance = Mahalanobis distance.



Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a blocking with MWBD $\leq 4\lambda$, though does much better than that in practice.

Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a blocking with MWBD $\leq 4\lambda$, though does much better than that in practice.
- Sketch of proof:
- Each vertex is at most a path of two edges away from a block center $\Longrightarrow$
  In the worst case: two vertices $\{i\}, \{j\}$ in the same block can be connected by a path of four edges in the bottleneck subgraph (two from vertex $\{i\}$ to the block center, two from the block center to vertex $\{j\}$).

Sketch of proof cont'd

- Each vertex is at most a path of two edges away from a block center $\Longrightarrow$
  In the worst case: two vertices $\{i\}, \{j\}$ in the same block can be connected by a path of four edges in the bottleneck subgraph (two from vertex $\{i\}$ to the block center, two from the block center to vertex $\{j\}$).
- In worst case: $(i, k_1), (k_1, k_2), (k_2, k_3), (k_3, j)$ is a path of four edges connecting $\{i\}$ to $\{j\}$.
- Each edge has distance of at most λ^- $\Longrightarrow$
  The distances along the edges satisfy:

$$d_{ik_1} + d_{k_1k_2} + d_{k_2k_3} + d_{k_3j} \leq 4\lambda^- \leq 4\lambda.$$

Sketch of proof cont'd

- Since distances satisfy the triangle inequality:

$$d_{ik} + d_{kj} \geq d_{ij}$$

it follows that

$$d_{ij} \leq d_{ik_1} + d_{k_1 k_2} + d_{k_2 k_3} + d_{k_3 j} \leq 4\lambda^{-} \leq 4\lambda.$$

Sketch of proof cont'd

- Since distances satisfy the triangle inequality:

$$d_{ik} + d_{kj} \geq d_{ij}$$

it follows that

$$d_{ij} \leq d_{ik_1} + d_{k_1 k_2} + d_{k_2 k_3} + d_{k_3 j} \leq 4\lambda^{-} \leq 4\lambda.$$

- That is, every edge joining two vertices within the same block has distance $\leq 4\lambda$.
- The maximum within-block distance of the approximately optimal blocking is $\leq 4\lambda$.
- QED

Future Work

- Apply graph partitioning techniques to other statistical problems (e.g. statistical clustering).
- Improve theoretic results of algorithm.

Bibliography I

D.S. Hochbaum and D.B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. Journal of the ACM (JACM), 33(3):533–550, 1986.

P.R. Rosenbaum. Optimal matching for observational studies. Journal of the American Statistical Association, 84(408): 1024–1032, 1989.